Введение в искусственный интеллект. Машинное обучение

Лекция 2. Непараметрические методы классификации и регрессии

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

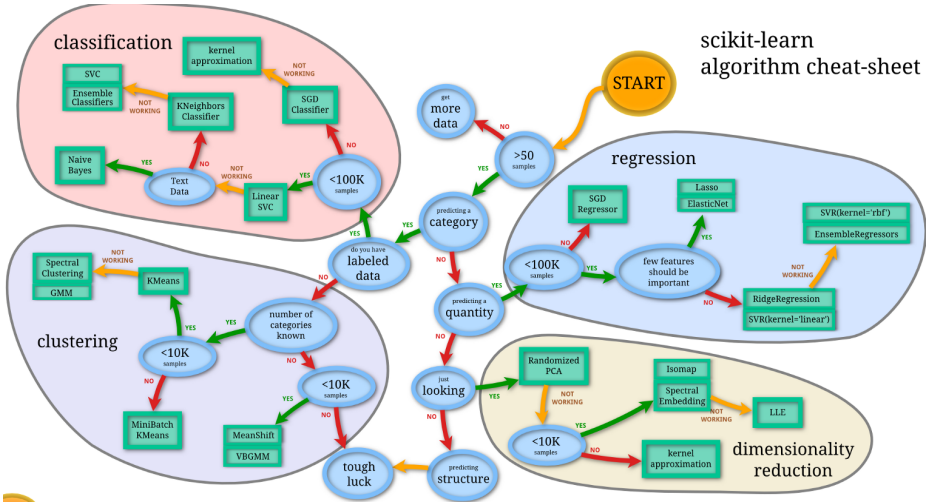
кафедра Математической Теории Интеллектуальных Систем

25 февраля 2020г.

План лекции

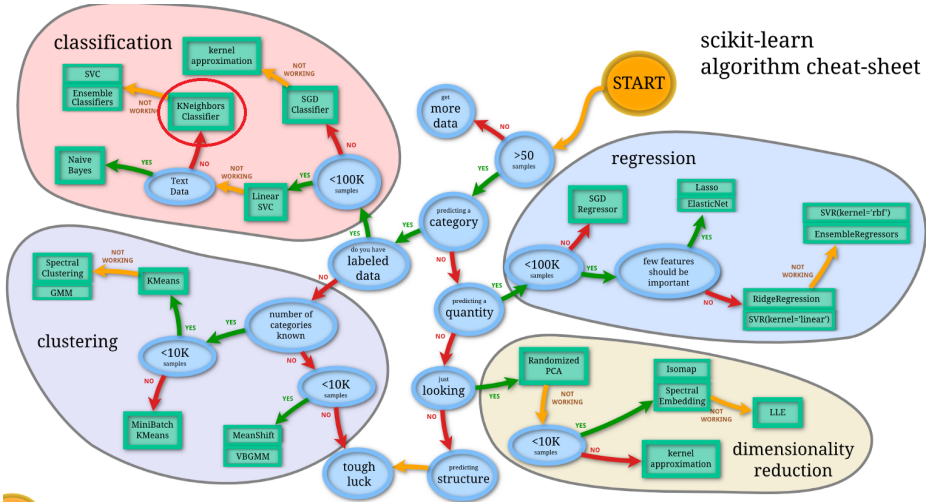
1. Метод ближайших соседей в задаче классификации
2. Непараметрическая регрессия
3. Методы поиска ближайшего соседа

Дорожная карта Scikit-Learn[1]



[1] https://scikit-learn.org/stable/tutorial/machine_learning_map/

Дорожная карта Scikit-Learn[1]



[1] https://scikit-learn.org/stable/tutorial/machine_learning_map/

Параметрические и непараметрические методы машинного обучения

Параметрические методы

- исходят из предположения, что искомая зависимость имеет некоторый специальный вид с точностью до некоторых параметров
- параметры находятся решением оптимизационной задачи

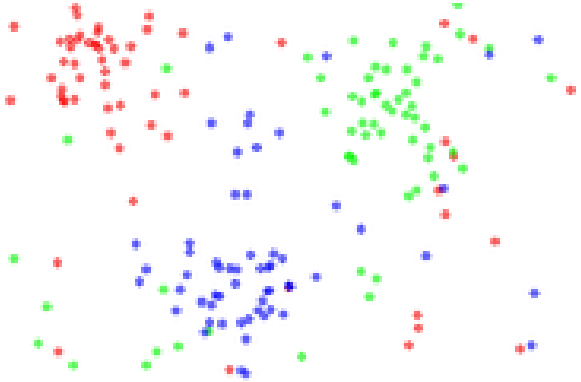
Непараметрические методы

Непараметрические методы – методы не являющиеся параметрическими

- Метрические алгоритмы, ядерные методы

Основное предположение

- "Близкие"объекты лежат в одном классе
- Близость задаётся метрикой
- Типичный пример [2]



[2] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Метод ближайшего соседа

- Параметр метода: метрика
- Алгоритм: по заданной метрике ищем ближайший объект в обучающей выборке и классифицируем объект так же

Преимущества

- Простота реализации (нет как таковой процедуры обучения в наивной реализации)
- Хорошая интерпретируемость

Недостатки

- Неустойчивость к выбросам
- Неоднозначность классификации при равных расстояниях до двух объектов
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- Не учитывается значение расстояния

Метод k ближайших соседей

- Параметр метода: метрика, **k**
- Алгоритм: по заданной метрике ищем k ближайших объектов в обучающей выборке и классифицируем объект как большинство из k объектов

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- Не учитывается значение расстояния

Метод k ближайших взвешенных соседей

- Параметры метода: метрика, k, **веса**
- Алгоритм: по заданной метрике ищем k ближайших объектов в обучающей выборке и классифицируем объект взвешенным голосованием

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- ~~Не учитывается значение расстояния~~

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса
  - Любая невозрастающая функция от порядкового номера
- Веса в зависимости от расстояния
  - Любая невозрастающая функция от расстояния
- Фиксированные веса объектов

Метод k ближайших взвешенных соседей среди набора эталонов

- Параметры метода: метрика, k, веса, **метод выбора эталонов**
- Алгоритм: по заданной метрике ищем k ближайших объектов среди эталонов выбранных из обучающей выборки и классифицируем объект взвешенным голосованием

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- ~~Необходимость хранить всю обучающую выборку~~
- ~~Алгоритм поиска вычислительно сложен~~
- ~~Не учитывается значение расстояния~~

Выбор эталонов

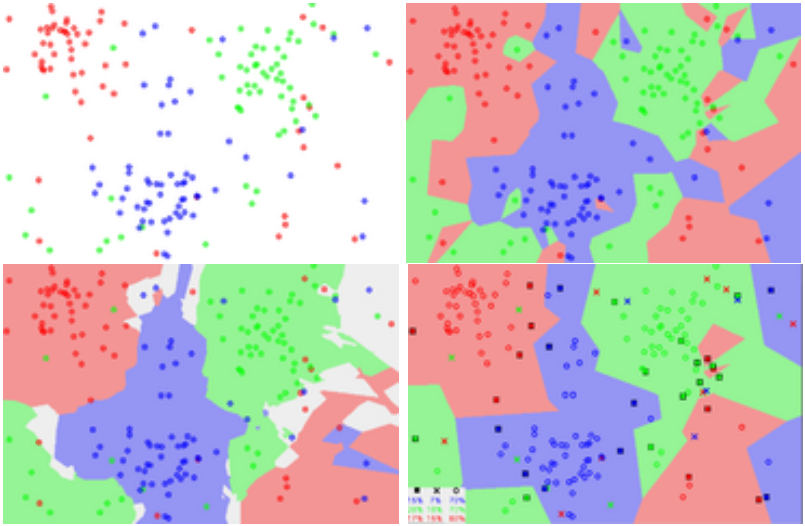
Задача

Получить примерно такое же качество работы алгоритма при меньшем количестве хранимых данных.
Возможно получить улучшение качества, так как в процессе выбора эталонов будут удалены выбросы.

Идеи

- Кластеризация объектов
- Жадный алгоритм

Примеры 1-nn, 5-nn, 1-nn с выбором эталонов



Реализация метода в scikit-learn

class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)

Основные параметры

- **n_neighbors** : int, optional (default = 5)
  Number of neighbors to use by default for kneighbors queries.
- **weights** : str or callable, optional (default = 'uniform')
  weight function used in prediction. Possible values:
  'uniform' : uniform weights 'distance' : weight points by the inverse of their distance.
  [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.
- **metric** : string or callable, default 'minkowski'
- **n_jobs** : int or None, optional (default=None)
  The number of parallel jobs to run for neighbors search

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода
  - Алгоритм подбора эталонов

Непараметрическая регрессия

- Главный минус параметрических моделей, что для описания зависимости необходимо иметь параметрическую модель
- В случае невозможности подбора адекватной модели имеет смысл пользоваться непараметрическими регрессионными методами

Предположение

Близким объектам соответствуют близкие ответы

Непараметрическая регрессия

Простейшая модель

Приближаем искомую зависимость константой в некоторой окрестности

Формула Надарая-Ватсона

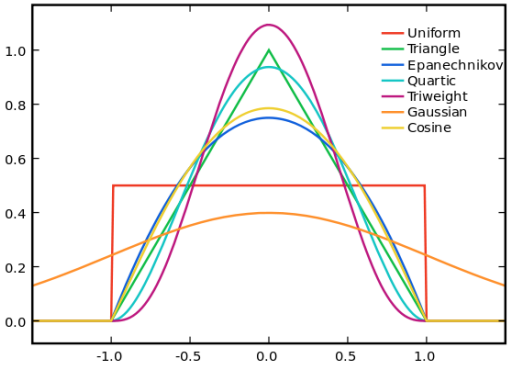
Если в окрестности точки несколько объектов из обучающей выборку, то разумно использовать взвешенное среднее в качестве предсказания алгоритма

$$a(x) = \frac{\sum\limits_i y_i \omega_i(x)}{\sum\limits_i \omega_i(x)},$$

где $\omega_i(x) = K_h(x, x_i)$, а функция K_h называется ядром с шириной окна сглаживания h.

Примеры ядер

- $K_h(x, x_i) = K(\frac{||x-x_i||}{h})$
- Типичные примеры [3]



[3] https://ru.wikipedia.org/wiki/Ядро_(статистика)

Bias-variance разложение в простейшем случае

$$E(a(x) - f(x))^2 = \left(f(x) - \frac{1}{k} \sum_{i=1}^{k} f(x_{(i)}) \right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- С ростом k разброс уменьшается
- А сдвиг увеличивается
- С ростом n сдвиг уменьшается

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода
  - Ширину окна сглаживания

Где могут быть полезны методы поиска ближайших соседей?

Методы поиска ближайших соседей

Точные

- Полный перебор
- K-мерное дерево (KD-tree) [4]
- Метрическое дерево (ball-tree) [5]

Приближенные

- Locality sensitive hashing (LSH)
- Navigable Small World (NSW)
- HNSW [6]

[4] Bentley, J. L. (1975). "Multidimensional binary search trees used for associative searching". Communications of the ACM. 18 (9): 509–517. doi:10.1145/361002.361007

[5] Omohundro, Stephen M. (1989) "Five Balltree Construction Algorithms"

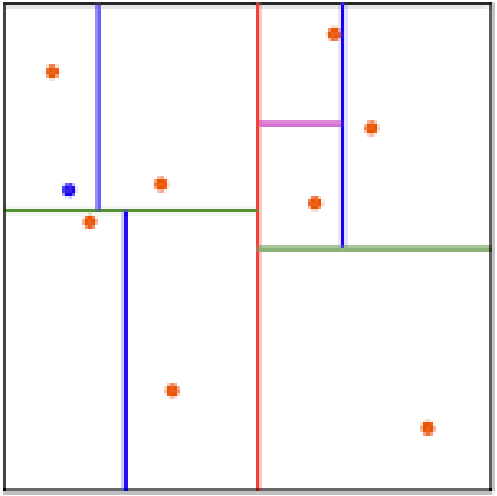
[6] `https://arxiv.org/abs/1603.09320`

K-мерное дерево

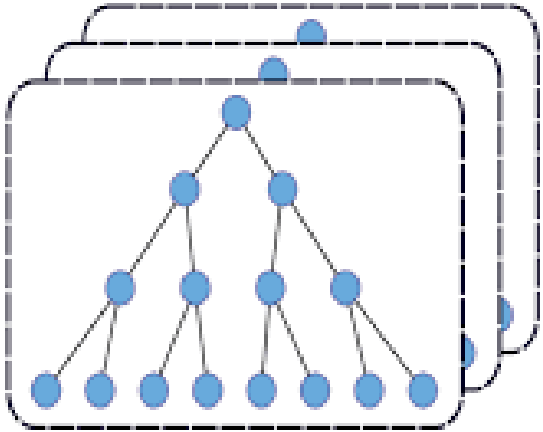
Алгоритм построения

1. Если количество элементов меньше некоторого порогового значение, то отбивается лист и в него помещаются все элементы, в противном случае переходим к следующему пункту
2. Случайно выбирается признак, по которому будет разделение. По этому признаку ищется медиана
3. Все объекты с выбранным признаком левее медианы идет в левое поддерево, остальные в правое
4. Для левого и правого поддерева применяется та же процедура построения

K-мерное дерево



Kd-Tree in 2D



Multiple Randomized Kd-Trees

Поиск ближайшего соседа в K-мерном дереве

Алгоритм поиска I

Для нашего запроса идём по дереву и в соотвествующем листе ищем нужное количество ближайших соседей

Идея

Если расстояние до дальнего ближайшего соседа меньше, чем расстояние до разделяющей гиперплоскости, то это означает, что во втором поддереве ближайших соседей нет.

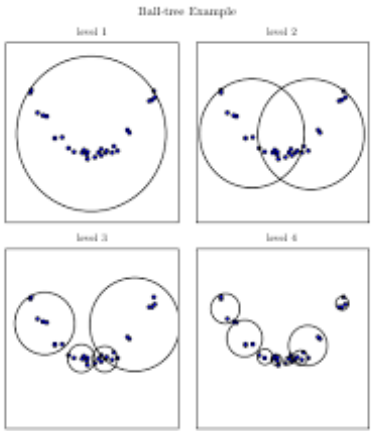
Алгоритм поиска II

1. Выполняем шаги алгоритма I, считая в каждой вершине расстояние до разделяющей гиперплоскости
2. Делаем обратный ход алгоритма, если расстояние до разделяющей гиперплоскости меньше, чем расстояние до дальнего ближайшего соседа

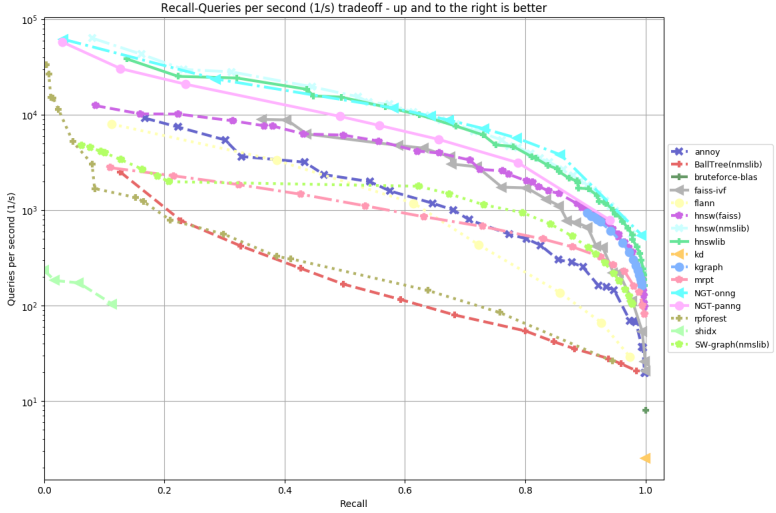
Метрическое дерево

Идея

Использовать вместо полугиперплоскостей шары



Сравнение методов поиска ближайших соседей [7]

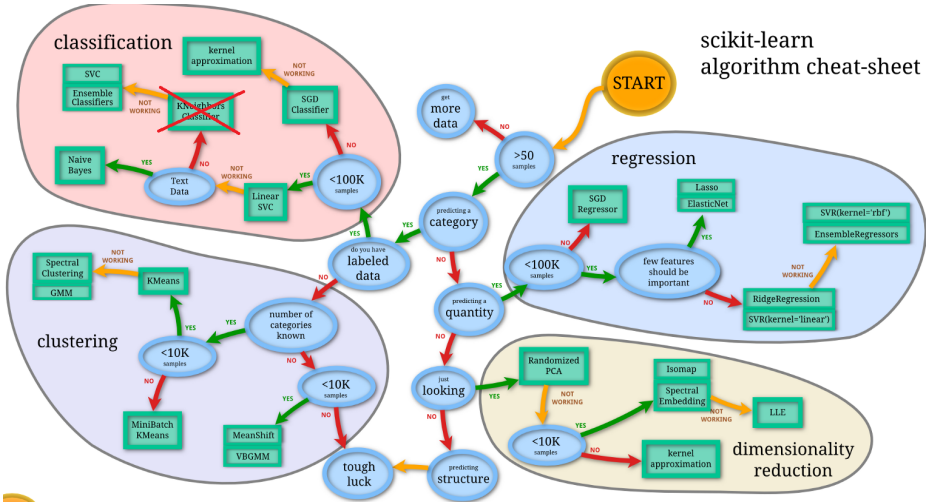


[7] https://github.com/erikbern/ann-benchmarks

Заключение

- Метод поиска ближайших соседей — важная задача теории алгоритмов
- Нужно помнить, что есть методы в среднем быстрее, чем полный перебор
- Для современных индустриальных систем характерно использование не точных, но очень быстрых алгоритмов поиска

Дорожная карта Scikit-Learn[8]



[8] https://scikit-learn.org/stable/tutorial/machine_learning_map/

Спасибо за внимание!